Global Science
The Journal of Research and Development

ISSN 2808-7259 (media online)

Implementasi Web Service pada Aplikasi Pemeriksaan Berkas Kelengkapan Penerbangan di Bandara Sam Ratulangi Manado Berbasis Web dan Android

[1]Green Welliam Manueke, [2]Standy Oei, [3]Waney Wanua Mamahit
[1]Dosen Jurusan Teknik Informatika Universitas Nusantara Manado
[2]Dosen Jurusan Teknik Informatika Universitas Nusantara Manado
[3]Mahasiswa Jurusan Teknik Informatika Universitas Nusantara Manado

Abstrak

Penelitian ini bertujuan untuk membantu proses pemeriksaan berkas kelengkapan keberangkatan di Bandara Sam Ratulangi Manado. Diharapkan dengan adanya sistem yang baru ini, berbasis Web dan Android, dapat mengatasi kendala dalam proses konfirmasi dan validasi berkas kelengkapan, terutama validasi berkas surat hasil pemeriksaaan rapid dan swab test. Menjadikan efisien waktu pada proses pemeriksaan menjadi lebih cepat.

Kata Kunci: ***web service, berkas penerbangan, bandara sam ratulangi***

I. PENDAHULUAN

Teknologi internet merupakan media informasi yang sangat bermanfaat untuk mendapatkan informasi berupa pengetahuan. Internet dapat diakses dimana saja dan kapan saja selama ada koneksi internet dari *provider* yang tersedia. Teknologi Internet sangat berdampak besar bagi kehidupan manusia sehingga memudahkan semua orang untuk mendapat informasi [1]. *Web Service* atau Web API salah satu teknologi yang bisa mengintegrasikan data lebih dari dua sistem yang berbeda.

Bandara Sam Ratulangi Manado merupakan bandar udara yang terletak di Sulawesi Utara dimana pemeriksaan berkas terutama surat hasil pemeriksaan test kesehatan dilakukan langsung di lokasi. Dalam masa pandemik ini dimana teknologi informasi sangat dibutuhkan, dalam hal ini pemeriksaan kelengkapan berkas keberangkatan di bandara Sam Ratulangi Manado dilakukan secara manual yang dimana penumpang harus menyerahkan berkas kelengkapan keberangkatan seperti identitas diri contohnya fotocopy KTP, tiket penerbangan, surat hasil pemeriksaan test kesehatan, dan beserta berkas lainnya yang diperlukan untuk divalidasi oleh petugas.

Dalam proses validasi berkas dimana penumpang menyerahkan berkas kelengkapan keberangkatan dan di validasi langsung di bandara oleh petugas dan validasi berkas oleh petugas bisa dilakukan dengan cara setiap penumpang akan memasukan data beserta berkas sesuai *form online* yang disediakan untuk mendapatkan *QR Code* dan diverifikasi langsung oleh petugas di pintu masuk keberangkatan dan di cap manual surat hasil pemeriksaan test kesehatan sedangkan penumpang lain harus menunggu sesuai antrian yang dimana antrian penumpang akan menjadi lebih panjang.

Dalam pemeriksaan berkas yang dimasukkan oleh petugas maupun penumpang secara langsung atau lewat form online dimana verifikasi dan konfirmasi data form maupun berkas seperti surat hasil pemeriksaan test kesehatan yang dimasukan membutuhkan waktu lama dan berpotensi pada efisiensi waktu keberangkatan penumpang sehingga mengakibatkan penumpang ketinggalan pesawat. Berdasarkan hasil pengamatan di Bandara Sam Ratulangi Manado permasalahan yang ada pada proses validasi berkas kelengkapan dapat diatasi dengan membangun suatu sistem yang bisa memvalidasi berkas terutama surat hasil pemeriksaan test kesehatan secara otomatis berdasarkan data yang tersedia di berbagai instansi kesehatan.

Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana agar penyerahan berkas kelengkapan untuk keberangkatan di bandara waktunya bisa lebih efisien sehingga mengurangi dampak keterlambatan penerbangan?

2. Seberapa efisien penggunaan *web service* dapat menghemat waktu verifikasi berkas?

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)

Tujuan Penelitian

Tujuan dari penelitian ini yaitu merancang dan membangun sistem yang bisa melakukan verifikasi berkas kelengkapan terutama surat hasil pemeriksaan test kesehatan secara otomatis dengan memanfaatkan teknologi *Web Service* atau *Web API*.

Manfaat Penelitian

Ada beberapa manfaat dari penulisan ini, yaitu:

1. Dapat menambah pengalaman bagi mahasiswa ketika secara langsung terlibat dalam pembuatan proyek yang sebenarnya.

2. Dapat menambah keterampilan bagi peneliti dalam merancang dan membangun sistem lintas platform dengan mengimplementasikan *Web Service* dan *Web API* serta menambah pengetahuan dalam menganalisis dan merancang *Database*.

II. LANDASAN TEORI

Sistem

Sistem merupakan kumpulan komponen yang memilki fungsi masing-masing dan terhubung satu sama lain untuk memenuhi suatu tujuan tertentu.

Sistem Informasi

Sistem informasi adalah serangkaian komponen berupa manusia, prosedur, data, dan teknologi (seperti komputer) yang digunakan untuk melakukan sebuah proses untuk menghasilkan informasi yang bernilai untuk pengambilan keputusan. Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang berguna dan berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata, dapat digunakan untuk pengambilan keputusan [2].

Web Service

Web Service adalah unit kode dikelola yang bisa dipanggil secara remote menggunakan HTTP. Artinya, bisa diaktifkan dengan menggunakan permintaan HTTP. Web Service memungkinkan Anda untuk mengekspos fungsi kode yang ada melalui jaringan. Setelah terekspos pada jaringan, aplikasi lain dapat menggunakan fungsi dari program Anda [3].

Web Server

Web Server adalah sebuah komputer yang terdiri dari perangkat keras dan perangkat lunak. Secara bentuk fisik dan cara kerjanya, perangkat keras *Web Server* tidak berbeda dngan komputer rumah atau PC, yang membedakan adalah kapasitas dan kapabilitasnya. Perbedaan tersebut dikarenakan *Web Server* bekerja sebagai penyedia layanan yang dapat diakses oleh banyak pengguna, sehingga dibutuhkan kapasitas dan kapabilitas yang besar dibandingkan PC [3].

HTTP

HTTP adalah singkatan dari *Hypertext Transfer Protocol* yang merupakan media transmisi untuk mengirimkan dokumen *hypertext*. HTTP pada versi pertama hanya bisa mengirimkan dokumen *hypertext* (dokumen yang berisi tautan ke dokumen lain). Seiring perkembangan HTTP sudah dapat digunakan ntuk mentransfer jenis file lain.

REST API

Nama *Representational State Transfer* (REST) diciptakan oleh Roy Fielding dari University of California. Kinerja, skalabilitas, kesederhanaan, portabilitas, dan kemampuan modifikasi adalah prinsip utama di balik desain REST [4]. REST API memungkinkan berbagai sistem untuk berkomunikasi dan mengirim/menerima data dengan cara yang sangat sederhana. Setiap dan setiap panggilan API REST memiliki hubungan antara kata kerja HTTP dan URL. Sumber daya dalam database dalam suatu aplikasi dapat dipetakan dengan titik akhir API di REST [4].

Beberapa perintah HTTP antara lain fungsi GET, POST, PUT, UPDATE atau DELETE. Supaya informasi yang diterima lebih mudah dibaca dan di-parsing disisi *client*, maka balasan yang dikirimkan oleh API Server dalam bentuk JSON [4].

RESTful *services* adalah teknologi yang bekerja menggunakan *resource* untuk membangun sistem terdistribusi, sehingga didesain dengan menekankan pada kesederhanaan, skalabilitas, serta kegunaan. *Web service* yang berorientasi pada *resource* dalam implementasinya banyak menggunakan REST, dengan menyediakan *resource-resource* sebagai layanan dan bukan kumpulan dari aktifitas yang mengolah *resource*. RESTful *Service* digunakan karena pada aplikasi *web client* dan *android client* akan mengolah *resource-resource* tersebut. Gaya arsitektur REST didasarkan pada permintaan dan pesan tanggapan yang ditransfer antara *client* dan *server* tanpa simpul

yang berpartisipasi melacak keadaan sesi sebelumnya [4].

Ada empat prinsip utama teknologi yang mendasari Metode REST [4], yaitu:
1. *Resource identifier* melalui *Uniform Resource Identifier* (URI), yaitu sekumpulan sumber daya yang interaksi antar *client* diindetifikasi oleh REST *Web service*.
2. *Uniform interface*, menggunakan operasi PUT, GET, POST dan DELETE untuk memanipulasi manipulasi CRUD (*Create*, *Read*, *Update*, *Delete*).
3. *Self-descriptive messages*, disamping *Metadata* dapat digunakan, sumber daya informasi tidak terikat, juga berbagai format konten (HTML, XML, PDF, JPEG, Plain Text dan lainnya) dapat diakses.
4. *Stateful interactions*, dapat berinteraksi dengan suatu sumber daya bersifat *stateless*, dengan *request messages* tergantung jenis kontennya melalui *hyperlink*.

Javascript

Javascript merupakan bahasa pemrograman untuk digunakan di sisi *client* maupun *server*. Javascript dirancang oleh Brendan Eich dalam waktu sepuluh hari pada bulan Mei 1995 untuk membuat halaman web berfungsi secara dinamis. Seiring perkembangan Javascript sudah bisa digunakan untuk tujuan umum [5].

JSON

JSON singkatan dari Javascript Object Notation. JSON merupakan format pertukaran data yang ringan, mudah di baca dan ditulis manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Penulisan format data menggunakan JSON sangatlah singkat karena menggunakan notasi-notasi javascript seperti kurung siku dan kurung kurawal [3]. Contoh Respon JSON bisa dilihat pada Gambar 1.

```
{
  "success":true,
  "message":"Data ditemukan",
  "data":{
    "nama_kendaraan":"Beat Merahku",
    "plat_nomor":"H-1123-VM",
    "tipe":"BeAt",
    "vendor":"Honda",
    "roda":2
  }
}
```


Gambar 1. Contoh Respon JSON

Database

Database adalah sekumpulan data tersebar yang berhubungan secara logis, dan penjelasan dari data ini dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi [6].

DBMS

DBMS singkatan dari *Database Management System*. DBMS merupakan perangkat lunak untuk mengendalikan pembuatan, pemeliharaan, pengolahan, dan penggunaan data yang berskala besar. Penggunaan DBMS saat ini merupakan hal yang sangat penting dalam segala aspek, baik itu dalam skala yang besar atau kecil [7].

MongoDB

MongoDB merupakan *database* NoSQL yang berorientasi dokumen dengan skema fleksibel dan bahasa query berbasis JSON. NoSQL merupakan singkatan dari "non-relational". NoSQL merupakan konsep query tanpa adanya relasi antara file yang biasanya disebut tabel pada konsep SQL tanpa mempedulikan konsistensi data yang kuat [8].

MVC

MVC merupakan singkatan dari *Model View Controller*. MVC merupakan konsep pengkodean aplikasi untuk memisahkan komponen-komponen pada kode program sebagai berikut [6]:
• *Model*, berisi kode program yang digunakan untuk memanipulasi *database*.
• *View*, berisi *template* HTML/XML/JSON atau PHP untuk menampilkan data pada *browser*.
• *Controller*, berisi kode program yang digunakan untuk mengontrol aliran aplikasi (sebagai pengontrol *Model* dan *View*).

PHP

PHP singkatan dari *Hypertext Preprocessor*. PHP merupakan bahasa pemrograman *server-side* yang dimana program akan dijalankan di *server* dan hasilnya akan diintegrasikan ke dalam kode sumber HTML [9].

Framework

Framework adalah kerangka kode yang dapat disempurnakan dengan *classes* yang spesifik atau dengan fungsi yang telah dirancang untuk mengatasi masalah yang dihadapi. Dapat disimpulkan bahwa *framework* biasanya bersifat *object–oriented* dan merupakan suatu desain sistem yang dapat digunakan kembali. Tujuannya untuk mengurangi pembuatan kembali kode yang sama

ISSN 2808-7259 (media online)

sehingga dapat lebih fokus mengerjakan bagian-bagian tertentu [10].

Laravel

Laravel adalah aplikasi *web* dengan *sintax* ekspresif dan elegan. Bertujuan untuk membuat proses *developing* yang menyenangkan tanpa mengorbankan fungsionalitas aplikasi pengembangan *web*. Laravel berusaha untuk mengurangi tugas umum yang digunakan dalam sebagian besar proyek *web*, seperti otentikasi, *routing*, sesi, dan *caching*. Laravel merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya [10].

QR Code

QR Code merupakan kode batang yang dikembangkan oleh Denso Wave, sebuah divisi Denso Corporation asal Jepang. Publikasi QR Code dilakukan pada 1994. Kode batang ini kemudian banyak sekali dipakai kerena fungsionalitas utamanya dapat menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Konten yang tersimpan pada QR Code dapat berupa teks huruf, angka, dan kode biner.

Umumnya, QR Code berisi alamat URL sebuah laman *web* atau iklan dan promosi produk komersial. QR Code dapat diterjemahkan menjadi kode respon cepat atau merupakan singkatan dari *Quick Response Code*.

QR Code dikembangkan oleh Denso Corporation, sebuah perusahaan Jepang yang banyak bergerak di bidang otomotif. QR Code ini dipublikasikan pada tahun 1994 dengan fungsionalitas utama yaitu dapat dibaca dengan mudah oleh pemindai (*scanner*).

Di dalam *website* resminya, Denso Wave Incorporated (2014) mengklaim QR Code mampu menangani berbagai macam tipe data seperti *numeric*, alphabet, kanji, kana, hiragana, *symbol*, biner, dan *control code* dan mampu meng-*encode* hingga 7.098 karakter dalam satu simbol, selain itu QR Code juga diklaim tahan terhadap kotoran, dan kerusakan [10]. Contoh QR Code bisa dilihat pada Gambar 2.

Gambar 2. Contoh QR Code

Android

Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi Linux [11]. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc. yang kemudian dibeli oleh Google pada tahun 2005. Dalam usaha mengembangkan Android, pada tahun 2007 dibentuklah Open Handset Alliance (OHA), sebuah konsorsium dari beberapa perusahaan, yaitu Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, dan TMobile dengan tujuan untuk mengembangkan standar terbuka untuk perangkat mobile.

Pada tanggal 9 Desember 2008, diumumkan bahwa 14 orang anggota baru akan bergabung dengan proyek Android, termasuk PacketVideo, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc.

Untuk membangun atau membuat aplikasi berbasis Android, terdapat dua cara. Pertama, kita harus memiliki perangkat telepon seluler yang berbasis Android langsung. Kedua, menggunakan emulator yang sudah disediakan oleh Google [12].

Android memiliki karakteristik yang terdiri dari empat karakteristik, yaitu:
1) Terbuka: Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Bersifat *Open Source*.
2) Semua aplikasi dibuat sama: Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (*third-party application*). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.
3) Memecahkan hambatan pada aplikasi: Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari *web* dengan data pada ponsel seseorang seperti kontak pengguna, kalender atau lokasi geografis.
4) Pengembangan aplikasi yang cepat dan mudah: Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan aplikasi yang semakin baik. Android memiliki sekumpulan *tools* yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)

III. METODOLOGI PENELITIAN

Objek Penelitian

Tempat yang menjadi objek penelitian dari penulis adalah Bandara Sam Ratulangi Manado dengan fokus penelitian pada operasional dalam pemeriksaan berkas kelengkapan keberangkatan.

Analisa Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk perkiraan kebutuhan sistem baru. Adapun analisis kebutuhan sistem meliputi:

1. Perangkat Lunak

Untuk menjalankan sistem ini diperlukan beberapa perangkat lunak yang dapat dilihat pada Tabel 1.

Tabel 1. Perangkat Lunak (*Software*)

Perangkat Lunak	**Versi**	**Keterangan**
Ubuntu	18.04.4	Sistem operasi server yang digunakan
Microsoft Windows Entreprise x64	10.0.17763.1577 (2018)	Sistem operasi yang digunakan dalam pengembangan
Android	6.0	Sistem operasi *mobile* yang digunakan dalam pengembangan
Apache HTTP *Server*	2.4.29	Untuk Web Server sehingga sistem bisa diakses oleh pengguna.
Hypertext Preprocessor (PHP)	7.3	Bahasa pemrograman untuk membangun sistem di sisi server.
Laravel	8.0	Kerangka kerja web untuk membangun web service.
React	16.2	Pustaka untuk membangun antarmuka aplikasi
		menggunakan bahasa pemrograman Javascript
React Native	0.63	Pustaka untuk membangun antarmuka aplikasi android menggunakan bahasa pemrograman Javascript
Expo SDK	39.0.3	Kombinasi alat pengembangan yang digunakan untuk membangun antarmuka aplikasi web dan android menggunakan pustaka React Native
Composer	2.0.1	Pengatur dependensi bahasa pemrograman PHP untuk menambahkan pustaka pihak ketiga
NPM (*Node Package Manager*)	6.14.4	Pengatur dependensi bahasa pemrograman Javascript untuk menambahkan pustaka pihak ketiga
MongoDB	4.4.1	Untuk implementasi database
Sublime Text	3.2.2 Build 3211	Kode editor untuk menuliskan kode program
Draw.io	13.0.3	Untuk perancangan diagram analisa seperti Flowchart, DFD dan ERD
Google Chrome	86.0.4240.193	Web browser untuk mengakses

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)

		antarmuka sistem berbasis web
Postman	7.34.0	Untuk uji coba *REST API* dalam pengembangan *Web Service*

2. Perangkat Keras

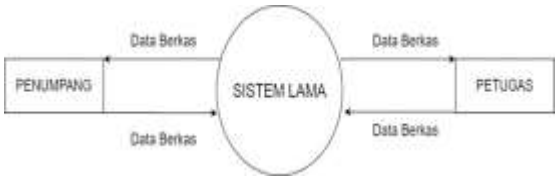
Untuk menjalankan sistem ini diperlukan beberapa perangkat keras yang dapat dilihat pada Tabel 2.

Tabel 2. Perangkat Keras (*Hardware*)

Perangkat Keras	**Spesifikasi Perangkat**
Pengembangan	
Prosesor	AMD E1-6010 Dual Core 1.35 GHz
Memori	4 GB
Hard Disk	500 GB
Prosesor (*Mobile*)	Spreadtrum SC7731 Quad Core 1.3 GHz
Memori (*Mobile* RAM)	1 GB
Penyimpanan (*Mobile* ROM)	8 GB
Produksi	
Prosesor	Intel(R) Xeon(R) Platinum 8163
Memori	1 GB
Hard Disk	39 GB

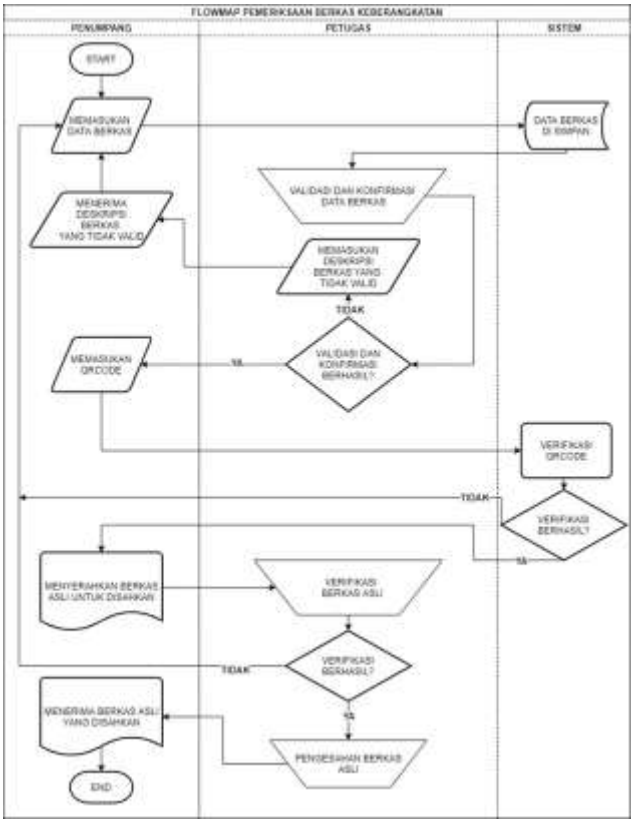
Tinjauan Sistem Lama

Secara garis besar keadaan operasional sistem lama dalam pemeriksaan berkas di Bandara Sam Ratulangi ini dapat digambarkan pada diagram berikut.



Gambar 3. Diagram Konteks Sistem Lama

Pada Gambar 3 merupakan gambaran sistem lama menggunakan diagram konteks yang memiliki dua entitas yaitu petugas dan penumpang. Data berkas menjadi objek yang ada pada sistem yang sedang dijalankan.



Gambar 4. *Flowmap* Diagram Sistem Lama

Pada Gambar 4 merupakan gambaran alur proses berkas keberangkatan pada sistem lama menggunakan *Flowmap* Diagram.

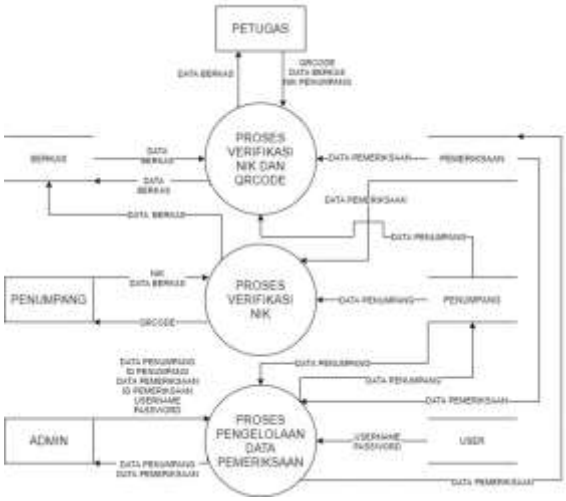
Berdasarkan diagram sistem lama yang sebelumnya dimana proses validasi dan konfirmasi berkas oleh petugas harus ada komunikasi langsung secara manual ke instansi kesehatan tempat penumpang melakukan pemeriksaan untuk memastikan apakah penumpang sudah melakukan pemeriksaan berdasarkan data pemeriksaan. Dari alur pemeriksaan dimana mengurangnya efisiensi waktu disaat pemeriksaan berkas ketika penumpang belum mengetahui *form* data kesehatan yang disediakan dan harus memasukan *form* data di bandara langsung yang mengakibatkan antrian akan menjadi lebih panjang.

Analisa dan Perancangan Sistem Baru

Gambar 5. Diagram Konteks Sistem Baru

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)

Gambar 5 di atas merupakan gambaran secara garis besar sistem yang akan dirancang menggunakan diagram konteks. Sistem yang dirancang memiliki tiga entitas yaitu Penumpang, Petugas, dan Admin.

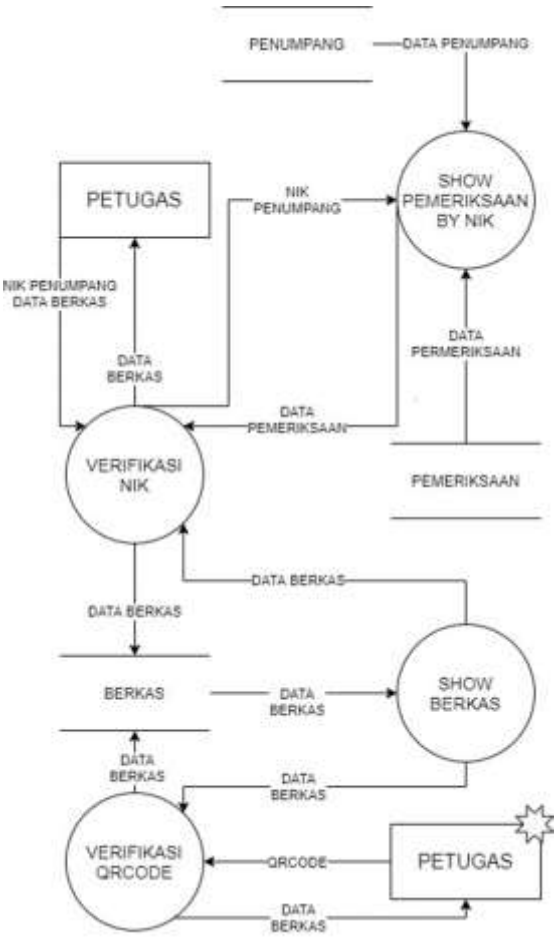


Gambar 6. *Data Flow Diagram* (DFD) Level 0

Gambar 6 di atas merupakan gambaran sistem baru yang memiliki 3 proses dan 4 penyimpanan data menggunakan DFD Level 0. Respon yang didapat oleh entitas Petugas saat memasukan NIK Penumpang merupakan data berkas yang dimana merupakan bagian dari data pemeriksaan.

Gambar 7. DFD Level 1 Proses Verifikasi NIK

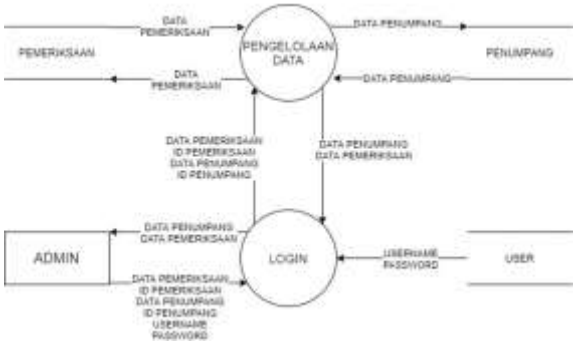
Gambar 7 di atas merupakan gambaran alir data dari Proses Verifikasi NIK untuk antarmuka entitas Penumpang yang memiliki 2 proses. Proses verifikasi berdasarkan data pemeriksaan yang dimana NIK sebagai parameter. NIK yang sebagai untuk mengetahui penumpang yang sudah melakukan pemeriksaan. Masa berlaku data pemeriksaan akan diperiksa pada proses verifikasi. Data berkas baru akan dimasukan saat proses verifikasi berhasil dan QR Code akan dihasilkan.



Gambar 8. DFD Level 1 Proses Verifikasi NIK dan QR CODE

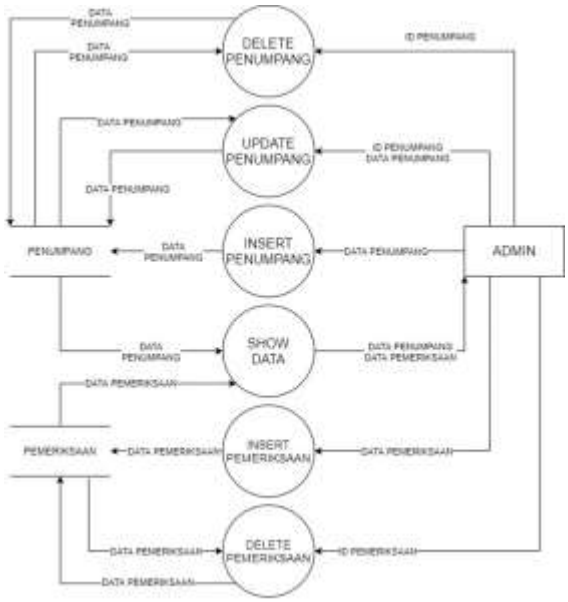
Gambar 8 di atas merupakan gambaran alir data dari Proses Verifikasi QR Code untuk antarmuka entitas Petugas memiliki 4 proses. QRCode dari Petugas didapat langsung dari Penumpang tanpa adanya proses dari sistem. Proses verifikasi NIK pada Petugas tidak mengembalikan QR CODE melainkan data berkas pada penumpang tertentu.

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)



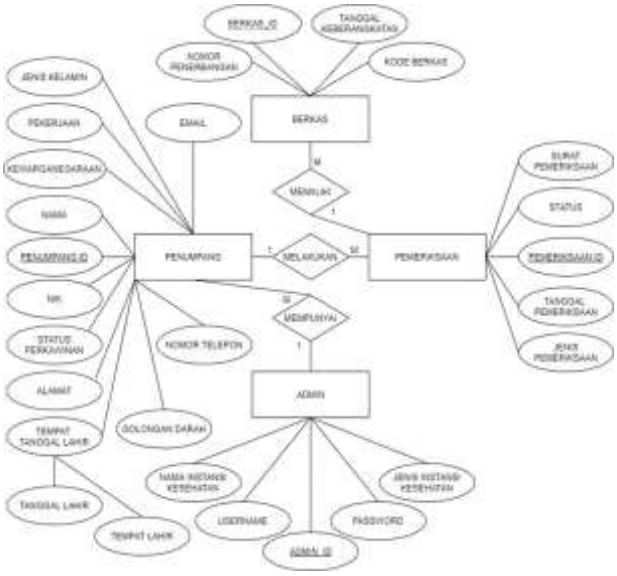
Gambar 9. DFD Level 1 Proses Pengelolaan Data Pemeriksaan

Gambar 9 di atas merupakan gambaran alir data dari Proses Pengelolaan Data untuk antarmuka entitas Admin memiliki 2 proses. Fitur Login dibuat untuk mengotentikasi Admin dari berbagai intansi kesehatan. Semua Admin dari berbagai instansi kesehatan mempunyai akses masing-masing untuk mengelola data pemeriksaan dan data penumpang yang melakukan pemeriksaan.



Gambar 10. DFD Level 2 Proses Pengelolaan Data

Gambar 10 di atas merupakan rincian proses Pengelolaan Data pada Diagram Alir Data Proses Pengelolaan Data Pemeriksaan Level 1 untuk antarmuka entitas Admin yang memiliki 6 proses. Admin bisa mengelola data pemeriksaan setiap penumpang berserta data identitas penumpang.



Gambar 11. Diagram Hubungan Entitas Sistem Baru

Gambar 11 di atas merupakan gambaran penyimpanan data pada sistem yang dikembangkan dalam bentuk ERD atau *Entity Relationship Diagram*. Dalam desain tersebut terdapat 4 entitas dimana entitas Admin memiliki relasi antara entitas Penumpang dan Pemeriksaan yang berarti bahwa setiap Admin dari setiap instansi kesehatan memiliki data penumpang dan pemeriksaan yang masing-masing berbeda. Entitas penumpang menggambarkan data penumpang yang merupakan identitas kependudukan dari penumpang yang melakukan pemeriksaan. Relasi *One-to-Many* pada entitas Pemeriksaan dan Berkas dimana penumpang bisa memiliki QR CODE lebih dari satu untuk mencegah ketika QR CODE yang dimiliki penumpang sebelumnya sudah tidak tersedia.

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini proses implementasi program ini merupakan bagian dari metode yang digunakan oleh penulis. Bahasa pemrograman yang digunakan penulis untuk membangun *Web Service* ini adalah PHP dan *framework* Laravel. Setelah *Web Service* selesai dibuat barulah dilakukan pembuatan tampilan antarmuka aplikasi berbasis *web* menggunakan bahasa pemrograman Javascript dengan *framework* React serta aplikasi berbasis android dengan menggunakan bahasa pemrograman Javascript dan framework React Native. Aplikasi android yang sebagai tampilan akan mengakses *endpoint* pemeriksaan untuk memeriksa kelengkapan berkas penumpang.

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)

Tampilan Halaman Form Login

Gambar 12. Tampilan Form Login

Pada halaman form login yang terdapat pada Gambar 12 digunakan untuk mengotentikasi setiap admin dari instansi kesehatan untuk melakukan pengelolaan data pemeriksaan setiap penumpang. Pada fitur login aplikasi hanya dibangun di *web*.

Tampilan Halaman Admin

Gambar 13. Tampilan Halaman Admin

Pada Gambar 13 halaman admin digunakan untuk melakukan pengelolaan data penumpang yang telah melakukan pemeriksaan. Admin dari instansi kesehatan bisa melihat, menambahkan, mengubah dan menghapus data penumpang yang melakukan pemeriksaan. Pada fitur admin hanya dibangun di *web*.

Tampilan Halaman Pemeriksaan

Gambar 14. Tampilan Halaman Pemeriksaan

Pada Gambar 14 halaman pemeriksaan digunakan untuk melakukan pengelohan data pemeriksaan pada penumpang tertentu yang telah dipilih oleh pengguna admin pada tampilan yang terdapat pada Gambar 13. Admin dari instansi kesehatan bisa melihat, menambahkan, menghapus, dan mencetak riwayat pemeriksaan pada penumpang tertentu.

Tampilan Form Tambah Pemeriksaan

Gambar 15. Tampilan Form Tambah Pemeriksaan

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)

Pada Gambar 15 tampilan form tambah pemeriksaan di halaman pemeriksaan digunakan untuk menambahkan data pemeriksaan saat hasil pemeriksaan pada penumpang tertentu sudah tersedia. Admin bisa mengunggah gambar surat hasil pemeriksaan. Tanggal pemeriksaan terisi secara otomatis sesuai tanggal dan waktu saat penambahan data pemeriksaan.

Tampilan Detail Pemeriksaan

Gambar 16. Tampilan Detail Pemeriksaan

Pada Gambar 16 tampilan detail pemeriksaan di halaman pemeriksaan digunakan untuk melihat gambar surat hasil pemeriksaan yang telah di unggah.

Tampilan Halaman Form Pengaturan Akun

Gambar 17. Tampilan Halaman Form Pengaturan Akun

Pada Gambar 17 tampilan halaman form pengaturan akun digunakan untuk mengubah nama instansi, jenis instansi, email dan kata sandi akun admin di setiap instansi kesehatan. Pada pengisian form kata sandi lama menjadi indikator untuk mengotentikasi admin dalam perubahan akun. Dengan mengklik *icon* pada salah satu *field* kata sandi dan kata sandi baru admin dapat melihat karakter asli dari kata sandi yang telah dimasukan.

Tampilan Halaman Form Data Keberangkatan

Gambar 18. Tampilan Halaman Form Data Keberangkatan

Pada Gambar 18 tampilan halaman form data keberangkatan penumpang melakukan verifikasi berkas pemeriksaan serta memeriksa masa berlaku pemeriksaan yang telah dilakukan dan QR Code akan dihasilkan setelah verifikasi berhasil selain itu maka sistem akan memberi tahu bahwa ada beberapa NIK yang tidak valid. Masa berlaku QR Code satu hari setelah tanggal keberangkatan. QR Code bisa di hasilkan berulang kali oleh penumpang. Satu QR Code bisa berlaku untuk beberapa NIK penumpang yang bisa diatur oleh penumpang yang sebagai pemandu.

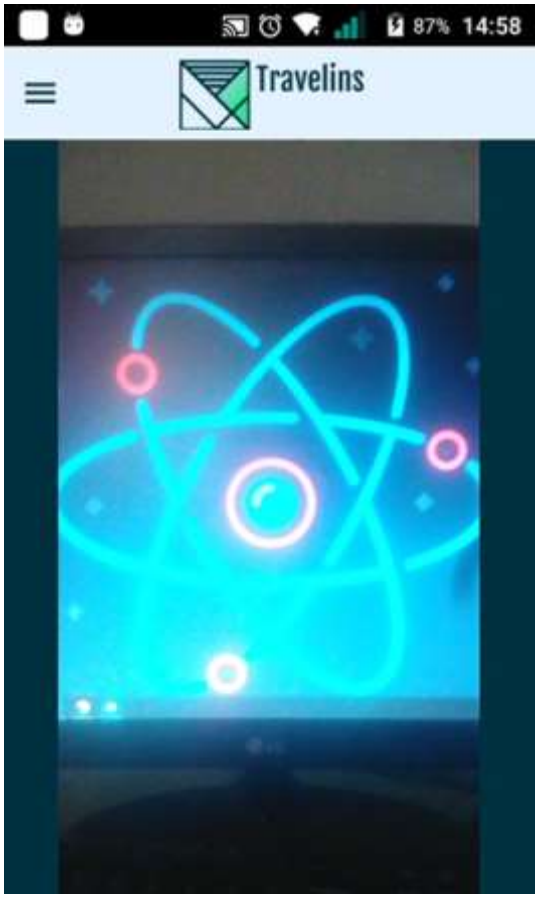
Tampilan Halaman QR Code Form Data Keberangkatan

Gambar 19. Tampilan Halaman QR Code Form Data Keberangkatan

Pada Gambar 19 tampilan halaman QR Code form data keberangkatan digunakan oleh penumpang yang sebagai pemandu untuk melihat dan menyimpan QR Code yang telah dihasilkan setelah memasukan data pada form yang terdapat pada Gambar 18.

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)

Antarmuka Pemindaian Kode Berkas



Gambar 20. Antarmuka Pemindaian Kode Berkas

Pada Gambar 20 antarmuka pemindaian kode berkas digunakan oleh petugas untuk memindai kode berkas yang telah didapatkan oleh penumpang dari form data keberangkatan yang terdapat pada Gambar 19 untuk verifikasi terakhir di bandara. Pada fitur pemindaian aplikasi hanya dibangun di android.

Antarmuka Form Data Keberangkatan

Gambar 21. Antarmuka Form Data Keberangkatan

Pada Gambar 21 antarmuka form data keberangkatan digunakan oleh petugas sebagai alternatif ketika salah satu penumpang tidak memiliki telepon genggam Android untuk verifikasi terakhir di bandara. Tanggal keberangkatan akan diisi secara otomatis oleh sistem. Pada form data keberangkatan pada Gambar 21 tidak menghasilkan QR Code melainkan data berkas yang terdiri dari data pemeriksaan dan data penumpang.

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)

Antarmuka Hasil Verifikasi

Gambar 22. Antarmuka Hasil Verifikasi

Pada Gambar 22 antarmuka hasil verifikasi digunakan oleh petugas untuk melihat daftar penumpang dalam satu keberangkatan setelah verifikasi terakhir berhasil dilakukan dari antarmuka pada Gambar 20 dan Gambar 21.

Antarmuka Detail Hasil Verifikasi

Gambar 23. Antarmuka Detail Hasil Verifikasi

Pada Gambar 23 antarmuka detail hasil verifikasi digunakan oleh petugas untuk melihat detail dari penumpang tertentu saat petugas memilih salah satu penumpang di daftar yang ada pada antarmuka pada Gambar 22.

URI /api/berkas (PATCH)

Gambar 24. Parameter URI /api/berkas (PATCH)

Global Science
The Journal of Research and Development
ISSN 2808-7259 (media online)

Pada Gambar 24 merupakan uji coba pada URI /api/berkas dengan method PATCH. Pada URI ini membutuhkan beberapa parameter yaitu api_token dan kode_berkas berupa format data JSON untuk melakukan verifikasi terakhir dari antarmuka yang dilakukan oleh petugas dari antarmuka yang terdapat pada Gambar 20.

Gambar 25. Respon URI /api/berkas (PATCH)

Pada Gambar 25 merupakan hasil uji coba pada URI /api/berkas dengan method PATCH. Pada URI ini menghasilkan data penumpang yang sudah terverifikasi berupa format data JSON dan ditampilkan pada antarmuka yang terdapat pada Gambar 22 dan Gambar 23.

URI /api/berkas (PUT)

Gambar 26. Parameter URI /api/berkas (PUT)

Pada Gambar 26 merupakan uji coba pada URI /api/berkas dengan method PUT. Pada URI ini membutuhkan beberapa parameter yaitu api_token, nomor_penerbangan, dan penumpang yang sudah terverifikasi berupa format data JSON untuk melakukan verifikasi terakhir dari antarmuka yang dilakukan oleh petugas dari antarmuka yang terdapat pada Gambar 21.

Gambar 27. Respon URI /api/berkas (PUT)

Pada Gambar 27 merupakan hasil uji coba pada URI /api/berkas dengan method PUT. Pada URI ini menghasilkan data penumpang yang sudah terverifikasi berupa format data JSON dan ditampilkan pada antarmuka yang terdapat pada Gambar 22 dan Gambar 23.

V. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil pembahasan yang dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. Dengan membangun sistem pemeriksaan berkas kelengkapan yang menerapkan teknologi *Web Service* maka dapat membantu petugas untuk memverifikasi penumpang langsung berdasarkan data pemeriksaan dari instansi kesehatan tertentu tempat penumpang melakukan pemeriksaan.
2. Dengan membangun sistem yang memiliki antarmuka yang disediakan dengan sekali pengisian dapat membantu penumpang dalam melakukan pemeriksaan berkas kelengkapan.

Saran

Berikut adalah saran-saran yang diharapkan dapat memperbaiki sistem ini, sehingga dalam perkembangannya di kemudian hari dapat menjadi lebih baik:

1. Dalam mengoperasikan sistem khususnya pengguna admin dari setiap instansi kesehatan harus diberikan pengetahuan dasar dalam mengoperasikan sistem.
2. Menambahkan fitur pemrosesan gambar untuk membantu pengguna admin dan penumpang dalam pengisian data.
3. Menambahkan fitur *backup* data pemeriksaan sebagai upaya menjamin keberlanjutan dan keamanan data.

ISSN 2808-7259 (media online)

DAFTAR PUSTAKA

[1] S. Oei, "IMPLEMENTASI IP CLOUD DAN DEMILITARIZED ZONE (DMZ) UNTUK PENGONTROLAN ROUTER JARAK JAUH," Prosiding Seminar Nasional Teknologi dan Sains (SNasTekS), pp. 223-234, September 2019.

[2] I. G. N. A. Widhiananda, I. M. A. Suyadnya dan K. O. Saputra, "RANCANG BANGUN APLIKASI RESERVASI SERVICE UNTUK BENGKEL SEPEDA MOTOR BERBASIS WEB," E-Journal SPEKTRUM, vol. 4, no. 2, pp. 97-104, Desember 2017.

[3] T. Choiri, N. Dengen dan I., "WEB SERVICE UNTUK WEB PROFIL SMP NEGERI 2 RANTEPAO TORAJA UTARA," vol. 2, no. 1, pp. 91-98, 2017.

[4] R. Choirudin dan A. Adil, "IMPLEMENTASI REST API WEB SERVICE DALAM MEMBANGUN APLIKASI MULTIPLATFORM UNTUK USAHA JASA," MATRIK, vol. 18, no. 2, pp. 284-293, Mei 2019.

[5] J. Dean, Web programming with HTML5, CSS, and JavaScript, Burlington: Jones & Bartlett Learning, 2019.

[6] U. M. Qibtiyah dan S. Rahayu, "Implementasi JSON Web Service pada Aplikasi Digital Library Politeknik Sukabumi," JTERA, vol. 2, no. 1, pp. 9-16, Juni 2017.

[7] E. F. W. Pramusinto dan S. Waluyo, "Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order," RESTI, vol. III, no. 2, pp. 106-112, 2019.

[8] V. Subramanian, Full Stack Web App Development with Mongo, Express, React, and Node, Bangalore: Apress Media, 2017.

[9] M. I. Perkasa dan E. B. Setiawan, "Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token," ULTIMA Computing, vol. 10, no. 1, pp. 19-26, Juni 2018.

[10] R. Akbar, M. Silvana dan A. F. Alizar, "Perancangan Aplikasi Pembayaran Non Tunai untuk Pengelolaan Bisnis Pencucian Mobil dengan Memanfaatkan Teknologi QR Code (Studi Kasus: Oto Pro Car Wash & Detailling Padang)," pp. 1-13, 16 Oktober 2019.

[11] A. Roihan, A. A. Wisanto, Y. Sulaeman, F. M. Nur, S. Williandi dan W. Pribadi, "Implementasi Metode Realtime, Live Data Dan Parsing JSON Berbasis Mobile Dengan Menggunakan Android Studio Dan PHP Native," Jurnal Teknologi Informasi, vol. 5, no. 2, pp. 117-123, Desember 2019.

[12] M. A. Mu'tashim, H. Anra dan H. Priyanto, "Sistem Layanan Pengaduan Masyarakat pada Balai Besar POM Kota Pontianak Berbasis Mobile," JUSTIN, vol. 8, no. 1, pp. 98-104, Januari 2020.